

UUU	UUU	EEEEEEEEEEEEEEEE	TTTTTTTTTTTTTT	PPPPPPPPPPPPPP
UUU	UUU	EEEEEEEEEEEEEEEE	TTTTTTTTTTTTTT	PPPPPPPPPPPPPP
UUU	UUU	EEEEEEEEEEEEEEEE	TTTTTTTTTTTTTT	PPPPPPPPPPPPPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEEEEEEEEEEE	TTT	PPPPPPPPPPPPPP
UUU	UUU	EEEEEEEEEEEE	TTT	PPPPPPPPPPPPPP
UUU	UUU	EEEEEEEEEEEE	TTT	PPPPPPPPPPPPPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEEEEEEEEEEE	TTT	PPP
UUU	UUU	EEEEEEEEEEEE	TTT	PPP
UUU	UUU	EEEEEEEEEEEE	TTT	PPP

RRRRRRRR	MM	MM	SSSSSSSS	TTTTTTTT	EEEEEEEEE	SSSSSSSS	TTTTTTTT	333333
RRRRRRRR	MM	MM	SSSSSSSS	TTTTTTTT	EEEEEEEEE	SSSSSSSS	TTTTTTTT	333333
RR	RR	MMMM	MMMM	SS	TT	EE	SS	33
RR	RR	MMMM	MMMM	SS	TT	EE	SS	33
RR	RR	MM	MM	SS	TT	EE	SS	33
RR	RR	MM	MM	SS	TT	EE	SS	33
RRRRRRRR	MM	MM	SSSSSS	TT	EEEEEEE	SSSSSS	TT	33
RRRRRRRR	MM	MM	SSSSSS	TT	EEEEEEE	SSSSSS	TT	33
RR	RR	MM	MM	SS	TT	EE	SS	33
RR	RR	MM	MM	SS	TT	EE	SS	33
RR	RR	MM	MM	SS	TT	EE	SS	33
RR	RR	MM	MM	SS	TT	EE	SS	33
RR	RR	MM	MM	SSSSSSSS	TT	EEEEEEEEE	SSSSSSSS	TT
RR	RR	MM	MM	SSSSSSSS	TT	EEEEEEEEE	SSSSSSSS	TT

LL		SSSSSSSS
LL		SSSSSSSS
LL		SS
LL		SS
LL		SS
LL		SSSSSS
LL		SSSSSS
LL		SS
LL		SS
LL		SS
LLLLLLLL		SSSSSSSS
LLLLLLLL		SSSSSSSS

```
0000 1 IDENT 'V04-000'
0000 74 $BEGIN RMSTEST3.009,--RMSTEST,<RELATIVE TEST PROGRAM>,<GBL,LONG>
0000 75
0000 76 :
0000 77
0000 78 .ENABL DBG
0000 79
0000 80 :
0000 81 : this program tests the relative file org
0000 82 :
0000 83 :
0000 84 :
0000 85 :
0000 86 : macros:
0000 87 :
0000 88
0000 89 .MACRO BUFF NAM,SIZE
0000 90 NAM'BUF':.
0000 91 .BLKB SIZE
0000 92 NAM'BSZ==SIZE
0000 93 .ENDM BUFF
0000 94
0000 95 :
0000 96 :
0000 97 :
0000 98
0000 99 .MACRO TYPE STRING, ?L
0000 100 STORE <STRING>
0000 101 BLBC VERTOSITY,L
0000 102 MOVL #SS.TMPX,CMDORAB+RAB$L,RBF
0000 103 MOVW #SS.TMPX1,CMDORAB+RAB$Q,RSZ
0000 104 SPUT RAB=CMDORAB,ERR=REPORT_ERROR
0000 105 BSBW ERR
0000 106 L: .ENDM TYPE
0000 107
0000 108
0000 109 :
0000 110
0000 111 .MACRO WTTYPE STRING
0000 112 SWAIT CMDORAB
0000 113 TYPE <STRING>
0000 114 .ENDM WTTYPE
0000 115
0000 116 :
0000 117
0000 118 .MACRO STORE STRING,PRE
0000 119 .SAVE
0000 120 .PSECT --$RMSNAM
0000 121 $$.TMPX=.
0000 122 PRE : store any carriage control info
0000 123 .ASCII %STRING%
0000 124 $$.TMPX1=-$$.TMPX
0000 125 .RESTORE
0000 126 .ENDM STORE
```

```
0000 128
0000 129 :
0000 130
0000 131 .MACRO FNM STRING
0000 132 STORE <STRING>
0000 133 MOVB #SS.TMPX1,FAB$B_FNS+RELATIVE_FAB
0000 134 MOVL #SS.TMPX,FAB$L_FNA+RELATIVE_FAB
0000 135 .ENDM
0000 136
0000 137 .MACRO BEGIN TSTNAM
0000 138 STORE <TSTNAM>
0000 139 MOVL #SS.TMPX,BEG_DESCR+4 : addr
0000 140 MOVL #SS.TMPX1,BEG_DESCR : len
0000 141 BSBW BEGPUT
0000 142 .ENDM BEGIN
0000 143 .MACRO FINISH TSTNAM
0000 144 STORE <TSTNAM>
0000 145 MOVL #SS.TMPX,FIN_DESCR+4 : addr
0000 146 MOVL #SS.TMPX1,FIN_DESCR : len
0000 147 BSBW FINPUT
0000 148 .ENDM FINISH
0000 149 .MACRO FIELD FLDNAM
0000 150 STORE <FLDNAM>
0000 151 MOVL #SS.TMPX,FLD_DESCR+4 : addr
0000 152 MOVL #SS.TMPX1,FLD_DESCR : len
0000 153 BSBW FLDPUT
0000 154 .ENDM FIELD
0000 155 .MACRO MBPT ?L
0000 156 BLBC VERBOSITY,L
0000 157 BPT
0000 158 L:
0000 159 .ENDM MBPT
0000 160
0000 161 :
0000 162
```

```

00000000 164 .PSECT RMSTEST.GBL, LONG
0000 165 .ALIGN LONG
0000 166 T3START:-
0000 167 RELATIVE_FAB:
0000 168 $FAB FAC=<GET,PUT,DEL,UPD>,-
0000 169 DNM=<TST$DISK:.FIL;1>,-
0000 170 NAM=NAMBLK,-
0000 171 FOP=<DFW,SUP>,-
0000 172 ORG=REL,-
0000 173 RAT=CR,-
0000 174 MRS=52,-
0000 175 MRN=500,-
0000 176 ALQ=0,-
0000 177 FSZ=4 : alq=0 forces extend
0050 178
0050 179 RELATIVE_RAB:
0050 180 $RAB FAB=RELATIVE_FAB,-
0050 181 UBF=RELBUF,-
0050 182 USZ=RELBSZ,-
0050 183 RBF=RELBUF,-
0050 184 RHB=HEAD,-
0050 185 MBF=3,-
0050 186 ROP=UIF
0094 187 ALLOC_XAB:
0094 188 $XABALL AID=0,-
0094 189 DEQ=4,-
0094 190 ALQ=4
0084 191 COUNTER:
00 0084 192 .BYTE 0
00 0085 193 COUNT2: .BYTE 0
00000000 0086 194 HEAD: .LONG 0
00000000 008A 195 KEY: .LONG 0
000000EE 008E 196 RFATBL: .BLKQ 6
00EE 197 .ALIGN LONG
00F0 198 BUFF REL,200
01B8 199 RFMSTR: .LONG RFML,RFMS
01C0 200 RFMS: .ASCII 'OPENED RELATIVE FILE WITH FILE ORG !AD !'
54 41 4C 45 52 20 44 45 4E 45 50 4F 000001C0'00000029'
54 49 57 20 45 4C 49 46 20 45 56 49 01CC
21 20 47 52 4F 20 45 4C 49 46 20 48 01D8
2F 21 20 44 41 01E4
00000029 01E9 201 RFML=-RFMS

```

000001C0'00000029' 01B8 199 RFMSTR: .LONG RFML,RFMS
 01C0 200 RFMS: .ASCII 'OPENED RELATIVE FILE WITH FILE ORG !AD !/'
 01CC
 01D8
 01E4
 00000029 01E9 201 RFML=-RFMS

				01E9	203	RMT\$TEST_3A::		
				01E9	204	:WORD	^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>	
				01EB	205	BEGIN	<RELATIVE TESTS>	
				0200	206	:		
				0200	207	:		
				0200	208	:VARIABLE		
				0200	209	:		
				0200	210			
				0200	211	TYPE	<TEST WITH VARIABLE RECORD FORMAT>	
				022F	212	FNM	<RELVAR>	
				023D	213	MOVB	#FAB\$C_VAR,FAB\$B_RF\$M+RELATIVE_FAB	
				0242	214	BSBW	REL_TEST	
				0245	215			
				0245	216	:		
				0245	217	:VFC		
				0245	218	:		
				0245	219			
				0245	220	TYPE	<TEST WITH VFC RECORD FORMAT>	
				0274	221	FNM	<RELVFC>	
				0282	222	\$FAB_STORE	FAB=RELATIVE_FAB,-	
				0282	223		RFM=VFC,-	
				0282	224		FSZ=#4: need to set fsz	
				028F	225	BISL2	#FAB\$M_CIF,FAB\$L_FOP(R0)	: do create-if
				0297	226	BICL2	#FAB\$M_SUP,FAB\$L_FOP(R0)	: not supercede
				0298	227	\$CREATE	FAB=RELATIVE_FAB,-	
				0298	228		ERR=REPORT_ERROR	
				02AC	229	BSBW	ERR	
				02AF	230	CMPL	R0,#RMSS_CREATED	: was it created
				02B6	231	BEQL	10\$	
				02B8	232	BISL2	#RAB\$M_UIF,-	: if opened, do updates
				02BA	233		RAB\$L_ROP+RELATIVE_RAB	
				02BD	234	10\$:	REL_TEST2	
				02C0	235	BSBW	#FAB\$M_CIF,-	
				02C6	236	BICL2	FAB\$L_FOP+RELATIVE_FAB	: restore bits
				02C9	237	BISL2	#FAB\$M_SUP,-	
				02CB	238		FAB\$L_FOP+RELATIVE_FAB	
				02CE	239			
				02CE	240			
				02CE	241	:	fix	
				02CE	242			
				02CE	243			
				02CE	244	TYPE	<TEST WITH FIXED LENGTH RECORDS>	
				02FD	245	FNM	<RELFIX>	
				030B	246	MOVB	#FAB\$C_FIX,FAB\$B_RF\$M+RELATIVE_FAB	
				0310	247	BSBW	REL_TEST	
				0313	248	CLRL	FAB\$L_XAB+RELATIVE_FAB	
				0317	249	FINISH	<RELATIVE TESTS>	
				032C	250			
				032D	251	RET		

```

032D 253 REL_TEST:
032D 254
032D 255 :
032D 256 :routine to put relative thru its paces, and call the locking routine
032D 257 :
032D 258
032D 259 SFAB_STORE FAB=RELATIVE_FAB,-
032D 260 ALQ=#0 ; make sure it's 0
032D 261 SCREATE FAB=R0,-
0335 262 ERR=REPORT_ERROR
0344 263 BSBW ERR
0347 264 $XABALL_STORE XAB=ALLOC_XAB,-
0347 265 ALQ=#4
0350 266 SFAB_STORE FAB=RELATIVE_FAB,-
0350 267 XAB=ALLOC_XAB
0358 268 SEXTEND FAB=R0,-
0358 269 BSBW ERR ; extend 4 blks, from alq
04 FC93' 30 036A 270 CMPL FABSL_STV+RELATIVE_FAB,#4 ; check returned stv
15 18 036D 271 BGEQ STVOK
0372 272
0374 273 STVOK: FIELD <STV ( NOT = ALLOC QTY, AFTER EXTEND)>
0389 274 CMPL XABSL_ALQ+ALLOC_XAB,#4 ; check alq
15 18 038E 275 BGEQ ALQOK
0390 276 FIELD <ALQ IN XAB ( NOT = ALLOC QTY, AFTER EXTEND)>
03A5 277 ALQOK: $CLOSE FAB=RELATIVE_FAB,-
03A5 278 BSBW ERR
03B6 279 $XABALL_STORE XAB=ALLOC_XAB,-
03B9 280 ALQ=#0
03C1 281 $OPEN FAB=RELATIVE_FAB,-
03C1 282 BSBW ERR
03D2 283 CMPL ALLOC_XAB+XABSL_ALQ,#5 ; alq=4 from extend + 1 since rel
05 FCCB' 30 03D5 284 BGEQ ALQOKT
15 18 03DA 285 FIELD <ALQ IN XAB ( NOT = ALLOC QTY, AFTER OPEN)>
03DC 286 CMPB RELATIVE_FAB+FABSB_RF, #FABSC_FIX
01 FC2A' 91 03F1 287 BEQL REL_TEST2
08 13 03F6 288 ALQOK1: $FAB_STORE FAB=RELATIVE_FAB,- ; on w/ it
03F8 289 XAB=#0 ; if var clear xab
03F8 290
03F8 291

```

```

0400 293 REL_TEST2:
0400 294
0400 295 :
0400 296 :entry point to bypass create
0400 297 :
0400 298 :
0400 299 SCONNECT RAB=RELATIVE_RAB,-
0400 300 BSBW ERR=REPORT_ERROR
FBEC' 30 0411 301
0414 302
0414 303 :
0414 304 :do 26 sequential puts with 'bad' data into file
0414 305 :then delete 5 of those records and do puts to put right data into them
0414 306 :update the other records and then do gets to make sure it's all ok
0414 307 :
0414 308 :
5B 59 01 00 0414 309 MOVL #1,R9 : r9 is record number
FC35 CF DE 0417 310 MOVAL RELATIVE_RAB,R11 : r11 is address of rab
041C 311 TYPE <FILL FILE>
044B 312 PUT_RECORD SEQ:::
FBCC 56 0F 01 044B 313 MOVL #15,R6 : all of length 15
CF 91 044E 314 CMPB #FABSC_FIX,FAB$B_RF+RELATIVE_FAB
03 12 0453 315 BNEQ 30$:
56 34 00 2C 0455 316 MOVL #52,R6 : len of fixed rec. is 52
FC90 CF 31 6E 00 0458 317 30$: MOVC5 #0,(SP),#^A/1/,R6,RELBUF: fill buffer for output
22 AB 56 B0 0460 318 MOVW R6,RABSW_RSZ(R11) : put ascii '1's into records
FC4D CF 59 D0 0464 319 MOVL R9,HEAD : give size of record
0469 320 :
0469 321 :
0469 322 :
0469 323 :seq mode is default
0469 324 :
0469 325 :
0469 326 SPUT RAB=R11,-
FB85' 30 0478 327 ERR=REPORT_ERROR
59 38 AB D1 047B 328 BSBW ERR
15 13 047F 329 CMPL RAB$L_BKT(R11),R9 : bkt should be rec. # on output
0481 330 BEQL BKT_OK
B1 59 1A F3 0496 331 FIELD <BKT IN RAB (RECORD NUMBER)>
0496 332 BKT_OK: AOBLEQ #26,R9,PUT_RECORD_SEQ : keep going?
049A 333 :
049A 334 :
049A 335 :check one record just to be sure
049A 336 :
049A 337 :
30 AB 59 0F 00 049A 338 MOVL #15,R9
FC19 CF DE 049D 339 MOVAL KEY,RAB$L_KBF(R11)
1E AB 01 90 04A3 340 MOVB #RABSC_KEY,RAB$B_RAC(R11)
FC0E CF 59 D0 04A7 341 MOVL R9,KEY
04AC 342 $GET RAB=R11,ERR=REPORT_ERROR
FB42' 30 04BB 343 BSBW ERR
01B7 30 04BE 344 BSBW CHK_BAD_DATA
04 AB 10 CA 04C1 345 BICL2 #RAB$M_OIF,RAB$L_ROP(R11) : in case its set
59 05 D0 04C5 346 MOVL #5,R9
04C8 347 FIND_DEL:
04C8 348 :
04C8 349 :

```

04C8 350 ;do some finds and deletes by keyed access
04C8 351 ;
04C8 352 ;
FBED CF 59 D0 04C8 353 MOVL R9,KEY
04CD 354 \$FIND RAB=R11,-
FB21' 30 04DC 355 BSBW ERR=REPORT_ERROR
04DF 356
04DF 357
04EE 359 BSBW ERR
59 05 C0 04F1 360 ADDL #5,R9
1A 59 D1 04F4 361 CMPL R9,#26
CF 19 04F7 362 BLSS FIND_DEL

04F9 364
 04F9 365 ;
 04F9 366 ;done with deletes, now 'put' into the deleted records correct data
 04F9 367 ;
 04F9 368
 59 05 D0 04F9 369 MOVL #5,R9
 04FC 370 PUT_RECORD_KEY:
 04FC 371
 04FC 372 ;
 04FC 373 ;try to get deleted records, hoping it fails
 04FC 374 ;then set nxr and get the deleted records and check them
 04FC 375 ;finally put the corrected (previously deleted)records
 04FC 376 ;
 04FC 377
 00C8 8F 00 6E 00 2C 04FC 378 MOVC5 #0,(SP),#0,#200,RELBUF ; clr relbuf, to make sure
 FBEA CF 0503
 FBAF CF 59 D0 0506 379 MOVL R9,KEY ; gets by key
 00000000'8F 50 D1 0514 380 \$GET RAB=R11 ; hope it fails
 1E 13 051B 381 CMPL R0,#RMSS\$_RNF ; record not found?
 5A 5B 50 D0 0532 382 BEQL ERR OK
 FAC8' 30 0535 383 FIELD <RETURNED ERROR CODE>
 FAC5' 30 0538 384 MOVL R11,R10
 00800000 8F C8 053B 385 BSBW REPORT_ERR
 FB10 CF 0541 386 BSBW ERR
 0544 387 ERR_OK: BISL #RAB\$M_NXR,- ; get non-existent record
 0544 388 RABSL_ROP+RELATIVE_RAB ; this should work
 FAAA' 30 0553 389 \$GET RAB=RT1,-
 011F 30 0556 390 ERR=REPORT_ERROR
 00800000 8F CA 0559 391 BSBW CHK_BAD_DATA ; check it out
 FAF2 CF 055F 392 BSBW #RAB\$M_NXR,- ; clear bit
 0222 30 0562 393 BICL RABSL_ROP+RELATIVE_RAB
 FB50 CF 59 D0 0565 394 BSBW SETUP ; fills buffer,rsz,head
 056A 395 MOVL R9,KEY ; keyed access
 056A 396 \$PUT RAB=R11,-
 FAB4' 30 0579 397 ERR=REPORT_ERROR
 59 05 C0 057C 398 BSBW ERR
 1A 59 D1 057F 400 ADDL #5,R9
 03 18 0582 401 CMPL R9,#26
 FF75 31 0584 402 BGEQ 10\$
 0587 403 BRW PUT_RECORD_KEY
 0587 404 10\$:
 0587 405
 0587 406 ;all done with that
 0587 407 ;
 0587 408 ;
 0587 409
 59 01 D0 0587 410 MOVL #1,R9
 058A 411
 058A 412 ;
 058A 413 ;do updates on all other records, by keyed access
 058A 414 ;
 058A 415 ;
 058A 416 UPDATE_RECORD:
 56 52 59 5A D4 058A 417 CLRL R10
 05 7B 058C 418 EDIV #5,R9,R2,R6
 56 D5 0591 419 TSTL R6 ; is it 5,10,15,20,25

F1 59 07 12 0593 420 BNEQ 20\$ #26, R9 UPDATE_RECORD : if so, skip it
 1A F3 0595 421 AOBLEQ #26, R9 UPDATE_RECORD : if so, skip it
 FB19 CF 0030 31 0599 422 BRW NO_MORE : all done
 59 D0 059C 423 20\$: MOVL R9-KEY
 05A1 424 SFIND RAB=R11,-
 FA4D' 30 05B0 425 BSBW ERR=REPORT_ERROR
 01D1 30 05B3 426 BSBW SETUP
 05B6 427 SUPDATE RAB=R11,-
 BE 59 FA38' 30 05C5 428 BSBW ERR=REPORT_ERROR : set up for put
 1A F3 05C8 429 BSBW ERR=REPORT_ERROR
 FA9D CF 00 90 05CC 430 BSBW ERR=REPORT_ERROR
 05D1 431 AOBLEQ #26, R9, UPDATE_RECORD
 NO_MORE: 432 NO_MORE:
 0600 433 MOVB #RAB\$C SEQ, RABSB_RAC+RELATIVE_RAB
 0600 434 TYPE <VERIFY CONTENTS OF FILE>
 0600 435 SREWIND RAB=R11,- : can now do gets
 F9EE' 30 060F 436 BSBW ERR=REPORT_ERROR
 00AA 30 0612 437 BSBW ERR
 0615 438 BSBW DO SOME_GETS
 0615 439 \$DISCONNECT RAB=R11,-
 F9D9' 30 0624 440 BSBW ERR=REPORT_ERROR
 0627 441 BSBW SCONNECT RAB=R11,-
 0627 442 BSBW ERR=REPORT_ERROR
 F9C7' 30 0636 443 BSBW ERR
 0083 30 0639 444 BSBW DO SOME_GETS
 063C 445 \$DISCONNECT RAB=R11,-
 063C 446 BSBW ERR=REPORT_ERROR
 F9B2' 30 064B 447 BSBW SCLOSE FAB=RELATIVE_FAB,-
 064E 448 BSBW ERR=REPORT_ERROR
 F99E' 30 065F 449 BSBW ERR
 0662 450 BSBW ERR
 F9CE CF 90 0662 451 BSBW ERR ; tell locking test which file
 00000034'EF 0666 452 MOVB FABSB_FNS+RELATIVE_FAB,-
 F9BD CF 00 0668 453 MOVL FABSB_FNS+LOCK_FAB
 0000002C'EF 066F 454 MOVL FABSL_FNA+RELATIVE_FAB,-
 0674 455 MOVL FABSL_FNA+LOCK_FAB
 F989' 30 0674 456 BSBW RMTTEST_SA : do locking tests
 05 0677 457
 0678 458 BSBW RSB
 0678 459
 0678 460
 0678 461 : subroutine to check 'bad' data (1st pass of puts)
 0678 462 : r11 is pointer to relative rab
 0678 463 : routine checks rsz and contents of record, now in buffer
 0678 464 :
 0678 465 :
 0678 466 :
 0678 467 CHK_BAD_DATA:
 56 0F D0 0678 468 MOVL #15, R6 : len of non-fixed records
 01 91 067B 469 CMPB #FABSC FIX,-
 F99F CF 067D 470 RELATIVE_FAB+FABSB_RFIM
 03 12 0680 471 BNEQ 10\$: if fix len is 52
 56 34 D0 0682 472 MOVL #52, R6
 22 AB 56 B1 0685 473 10\$: CMPW R6, RABSW_RSZ(R11) : check rsz
 15 13 0689 474 BEUL RSZ OK
 068B 475 FIELD <RSZ>
 06A0 476 RSZ_OK:

RMSTEST3
009

RELATIVE TEST PROGRAM :

F 8

16-SEP-1984 01:47:03 VAX/VMS Macro V04-00
5-SEP-1984 04:21:48 [UETP.SRC]RMSTEST3.MAR;1

Page 10
(9)

6E 00 31 28 BB 56	2D 06A0 477	CMPC5	R6,0RABSL_RBF(R11),#^A/1/,#0,(SP)
15	13 06A7 478	BEQL	REC OK
	06A9 479	FIELD	<RECORD>
	06BE 480 REC_OK:		
	05 06BE 481	RSB	

RM
00

DO_SOME_GETS::								
54	F9FB	CF	DE	06BF	483	MOVAL	RFATBL, R4	: r4 is index to rfatbl
	F9EC	CF	94	06C4	484	CLRB	COUNTER	: record number
	F9E9	CF	94	06C8	485	CLRB	COUNT2	
	57	1A	DO	06CC	486	MOVL	#26, R7	: 1st pass-r7 is # of rec.
5B	F97D	CF	DE	06CF	487	MOVAL	RELATIVE RAB, R11	: pointer to rab
1E	AB	00	90	06D4	488	MOVB	#RABSC SEQ, RABSB_RAC(R11)	: do all sequential gets
	00E1		30	06D8	489	BSBW	GET RECORD SEQ	
1E	AB	02	90	06DB	490	MOVB	#RABSC_RFA, RABSB_RAC(R11)	: do some gets by rfa
54	F9DB	CF	DE	06DF	491	MOVAL	RFATBL, R4	
	0180		30	06E4	492	BSBW	GET RECORD RFA	
1E	AB	01	90	06E7	493	MOVB	#RABSC KEY, RABSB_RAC(R11)	: do some gets by key
	01A2		30	06EB	494	BSBW	GET RECORD KEY	
1E	AB	00	90	06EE	495	MOVB	#RABSC_SEQ, RABSB_RAC(R11)	: do some seq gets
	57	0D	DO	06F2	496	MOVL	#13, R7	: 2nd pass-r7 is # of rec.
54	F9DD	CF	DE	06F5	497	MOVAL	RFATBL+24, R4	: 24=8.*3, starting in the middle
F9BS	CF	0D	90	06FA	498	MOVB	#13, COUNTER	ditto
F9B2	CF	94	06FF	499	CLRB	COUNT2		
	00B6		30	0703	500	BSBW	GET_RECORD_SEQ	
			05	0706	501			
					502	RSB		

		0707	504							
		0707	505	; subroutines to do gets and checks						
		0707	506							
		0707	507	; ;						
		0707	508	; ;						
		0707	509	CHECK_REC:						
		0707	510	;						
		0707	511	;						
		0707	512	; r9 is the record number						
		0707	513	as before r11 is the addr of the rab						
		0707	514	;						
		0707	515							
		F913 CF	03	91	0707	516	CMPB	#FABSC_VFC,FABSB_RFH+RELATIVE_FAB		
		18	12	070C	517	BNEQ	NO HEADER			
		2C BB	59	D1	070E	518	100\$:	CMPL		
		15	13	0712	519	BEQL	R9,DRABSL_RHB(R11) : compare header			
				0714	520	FIELD	NO HEADER			
				0729	521	NO_HEADER:	<HEADER OF A VFC RECORD>			
		1A	59	D1	0729	522	CMPL	R9,#26 ; is it last record?		
		05	05	12	072C	523	BNEQ	10\$		
		56	1A	D0	072E	524	MOVL	#26,R6 ; length is 26		
		07	1A	11	0731	525	BRB	20\$		
		5A	5A	D4	0733	526	10\$:	CLRL	R10	
		52	59	1A	0735	527	EDIV	#26,R9,R2,R6	; quadword divide	
55	56	00000040	8F	C1	073A	528	20\$:	ADDL3	#^A/A/-1,R6,R5	; r6 = r9 mod 26, rec. size
		F8D8 CF	01	91	0742	529	CMPB	#FABSC_FIX,FABSB_RFH+RELATIVE_FAB		
		03	12	0747	530	BNEQ	GOT_RS			
		56	34	D0	0749	531	MOVL	#52,R6	; mrs for fixed is 26	
		22 AB	56	B1	074C	532	GOT_RS:	CMPW	R6,DRABSW_RSZ(R11)	; check rsz
			15	13	0750	533	BEQL	OK_RSZ		
				0752	534	FIELD	<RSZ FIELD IN RAB>			
				0767	535					
		6E 00 55 28 BB	56 01	2D	0767	536	OK_RSZ:	CMPC5	R6,DRABSL_RBF(R11),R5,NO,(SP)	
				12	076E	537	10\$:	BNEQ	BADREC	
				05	0770	538		RSB	: return	
				0771	539			FIELD	<RECORD>	
				0771	540			RSB		
				05	0786	541				
						542				

0787 544 SETUP:
 0787 545
 0787 546:
 0787 547: routine to do setup for puts of correct data
 0787 548: r9 is record number -- input
 0787 549: output -- relbuf is filled in with correct char
 0787 550: -- head is filled in, and rsz is also
 0787 551:
 0787 552:
 1A 59 D1 0787 553 CMPL R9 #26
 05 05 12 078A 554 BNEQ 106
 56 1A D0 078C 555 MOVL #26,R6 ; len of last rec. is 26
 07 11 078F 556 BRB 20\$
 5A D4 0791 557 105: CLRL R10
 55 56 52 59 1A 78 0793 558 EDIV #26,R9,R2,R6
 00000040 8F C1 0798 559 205: ADDL3 #^A/A/-1,R6,R5 ; r6 is rec. # mod 26
 01 91 07A0 560 CMPB #FABSC FIX- ; r5 is char. to fill buffer
 F87A CF 03 12 07A2 561 FABSB_RFH+RELATIVE_FAB
 03 12 07A5 562 BNEQ 30\$
 F93E CF 56 55 56 34 D0 07A7 563 MOVL #52,R6 ; len. of fixed rec. is 52
 22 AB 6E 00 2C 07AA 564 305: MOVC5 #0,(SP),R5,R6,RELBUF ; fill relbuf
 F8FB CF 59 D0 07B2 565 MOVW R6,RABSW_RSZ(R11) ; fill rsz
 05 07B6 566 MOVL R9,HEAD ; fill header. in case vfc
 05 07B8 567 RSB

07BC 569
 07BC 570 GET_RECORD_SEQ:
 07BC 571
 07BC 572 :
 07BC 573 :
 07BC 574 :
 07BC 575 :
 00000000'8F 50 D1 07C5 576 SGET R11
 0F 12 07CC 577 CMPL R0, #RMSS_EOF
 57 F8E3 CF 91 07CE 578 BNEQ MORE
 01 12 07D3 579 CMPB COUNT2, R7
 05 07D5 580 BNEQ BADNR
 5A 5B, 581 RSB
 F824' D0 07D6 582 BADNR:
 30 07D9 583 MOVL R11, R10
 05 07DC 584 BSBW EOFPUT
 09 50 E8 07DD 585 RSB
 5A 5B, 586 MORE:
 F81A' 50 07E0 587 BLBS R0, 10\$
 F817' 50 07E3 588 MOVL R11, R10
 57 F8C8 CF 91 07E9 589 BSBW REPORT_ERR
 02 15 07EE 590 BSBW ERR
 E4 11 07F0 591 10\$: CMPB COUNT2, R7
 F8BE CF 96 07F2 592 BLEQ 20\$
 F8BB CF 96 07F6 593 20\$: BRB BADNR
 59 F8B6 CF 9A 07FA 594 INCB COUNTER
 38 AB 59 D1 07FF 595 INCB COUNT2
 15 13 0803 596 MOVZBL COUNTER, R9
 0805 597 CMPL R9, RABSL_BKT(R11)
 081A 598 BEQL RNOK
 600 RNOK: FIELD < BKT FIELD IN RAB >
 FEEA 30 081A 600 RNOK:
 5A D4 081D 601 BSBW CHECK_REC
 50 78 081F 602 CLRL R10 : quad word divide
 01 52 0824 603 EDIV #5, R9, R0, R2
 03 13 0827 604 CMPL R2, #1
 FF90 31 0829 605 BEQL SAV_RFA
 082C 606 BRW GET_RECORD_SEQ : continue
 082C 607 SAV_RFA:
 082C 608 :
 082C 609 : save record numbers 1, 6, 11, 16, 21, 26 on 1st pass
 082C 610 : check record numbers 16, 21, 26 on 2nd pass
 082C 611 :
 082C 612 :
 082C 613 :
 64 10 AB 2A 57 E9 082C 614 BLBC R7, SAV : which pass?
 06 29 082F 615 CMPC3 #6, RABSW_RFA(R11), (R4) : 2nd, check them
 1D 13 0834 616 BEQL RFÄ OK
 0836 617 FIELD <RFÄ>
 0848 618 MBPT
 54 08 C0 0853 619 RFA_OK:
 FF63 31 0853 620 ADDL2 #8, R4
 0856 621 BRW GET_RECORD_SEQ : on to next record
 84 10 AB 70 0859 622 SAV: MOVQ RABSW_RFA(R11), (R4)+ : 1st pass, save them
 52 54 01 C3 0850 623 SUBL3 #1, R4-R2 : also store record number
 62 59 90 0861 624 MOVB R9, (R2)

FF55 31 0864 626 BRW GET_RECORD_SEQ

0867 627
0867 628 GET_RECORD_RFA:

0867 629
0867 630 :
0867 631 :
0867 632 :
0867 633 :
0867 634 :
0867 635 :
0867 636 :
0867 637 :
0867 638 :
0867 639 :
0867 640 :
0867 641 :
088F 642 END_OF_RFA:
088F 643 :
0890 644 :
0890 645 GET_RECORD_KEY:
0890 646 :
0890 647 :
0890 648 :
0890 649 :
0890 650 :
0890 651 :
0896 652 :
0898 653 :
089E 654 :
08A3 655 :
08A6 656 :
08AB 657 :
08AE 658 :
08AE 659 GETCHK:
08AE 660 :
08AE 661 :
08BD 662 :
08C0 663 :
08C5 664 :
08C8 665 :
30 AB F826 CF DE 0890 651 MOVAL KEY,RABSL_KBF(R11) :
F81F CF 01 D0 0896 652 MOVL #1,KEY :
0010 30 0898 653 BSBW GETCHK :
F817 CF 1A D0 089E 654 MOVL #26,KEY :
0008 30 08A3 655 BSBW GETCHK :
F80F CF 0D D0 08A6 656 MOVL #13,KEY :
0000 31 08AB 657 BRW GETCHK :
08AE 658 :
08AE 659 GETCHK:
08AE 660 :
08AE 661 :
08BD 662 :
08C0 663 :
08C5 664 :
08C8 665 :
59 F740' 30 08BD 662 :
F7F6 CF D0 08C0 663 :
FE3F 31 08C5 664 :
08C8 665 :
5-SEP-1984 04:21:48 [UETP.SRC]RMSTEST3.MAR;1

CMPL R4,#RFATBL+48 : more rfa entries
BGEQ END_OF_RFA
MOVQ (R4)+ RABSW,RFA(R11) : Load rab w/ rfa
SGET RAB=R11,ERR=REPORT_ERROR
BSBW ERR
MOVBL RABSW-RFA+7(R11),R9 : get record number, as stored
BSBW CHECK_REC : r9 is now rec. #
BRB GET_RECORD_RFA

KEY,RABSL_KBF(R11) : get 1st record
#1,KEY : get and check
#26,KEY
#13,KEY
GETCHK
RAB=R11,-
ERR=REPORT_ERROR
ERR
KEY,R9
CHECK_REC

.END

\$\$._PSECT_EP	= 00000000		FAB\$V_FILE_MODE	= 00000004	D
\$\$._TAB	= 00000094	R D 01	FAB\$V_GET	= 00000001	D
\$\$._TABEND	= 00000084	R D 01	FAB\$V_LNM_MODE	= 00000000	D
\$\$._TMP	= 00000002	D	FAB\$V_PUT	= 00000000	D
\$\$._TMP1	= 00000002	D	FAB\$V_SUP	= 00000002	D
\$\$._TMP2	= 0000005B		FAB\$V_UPD	= 00000003	D
\$\$._TMPX	= 00000191	R D 04	FAB\$W_GBC	= 00000048	D
\$\$._TMPX1	= 00000003	D	FIND_DEL	000004C8 R D 01	
SSRMSTEST	= 0000001E		FINPUT	***** X 01	
SSRMS_PBUGCHK	= 00000010		FIN_DESCR	***** X 01	
SSRMS_TBUGCHK	= 00000008		FLDPUT	***** X 01	
SSRMS_UMODE	= 00000004		FLD_DESCR	***** X 01	
.AFLG	= 00000000	D	GETCHK	000008AE R D 01	
.FLG	= 00000001	D	GET_RECORD_KEY	00000890 R D 01	
.MOD	= 00000001	D	GET_RECORD_RFA	00000867 R D 01	
.TYP	= 000000CF		GET_RECORD_SEQ	000007BC R D 01	
.LEN	= 00000004	D	GOT_RS	0000074C R D 01	
ALLOC_XAB	00000094	R D 01	HEAD	000000B6 R D 01	
ALQOK	000003A5	R D 01	KEY	000000BA R D 01	
ALQOK1	000003F1	R D 01	LOCK_FAB	***** X 01	
BADNR	000007D6	R D 01	MORE	000007DD R D 01	
BADREC	00000771	R D 01	NAMBLK	***** X 01	
BEGPUT	***** X 01		NO_HEADER	00000729 R D 01	
BEG_DESCR	***** X 01		NO_MORE	000005CC R D 01	
BKT_OK	00000496	R D 01	OK_RSZ	00000767 R D 01	
CHECK_REC	00000707	R D 01	PUT_RECORD_KEY	000004FC R D 01	
CHK_BAD_DATA	00000678	R D 01	PUT_RECORD_SEQ	0000044B RG D 01	
CMD\$RAB	***** X 01		RAB\$B_RAC	= 0000001E D	
COUNT2	000000B5	R D 01	RAB\$C_BID	= 00000001 D	
COUNTER	000000B4	R D 01	RAB\$C_BLN	= 00000044 D	
DO SOME GETS	000006BF	RG D 01	RAB\$C_KEY	= 00000001 D	
END_OF_RFA	0000088F	R D 01	RAB\$C_RFA	= 00000002 D	
EOFPUT	***** X 01		RAB\$C_SEQ	= 00000000 D	
ERR	***** X 01		RAB\$L_BKT	= 00000038 D	
ERR_OK	00000538	R D 01	RAB\$L_CTX	= 00000018 D	
FAB\$B_DNS	= 00000035	D	RAB\$L_KBF	= 00000030 D	
FAB\$B_FNS	= 00000034	D	RAB\$L_RBF	= 00000028 D	
FAB\$B_FSZ	= 0000003F	D	RAB\$L_RHB	= 0000002C D	
FAB\$B_RFM	= 0000001F	D	RAB\$L_ROP	= 00000004 D	
FAB\$C_BID	= 00000003	D	RAB\$M_NXR	= 00800000 D	
FAB\$C_BLN	= 00000050	D	RAB\$M_UIF	= 00000010 D	
FAB\$C_FIX	= 00000001	D	RAB\$V_UIF	= 00000004 D	
FAB\$C_REL	= 00000010	D	RAB\$W_RFA	= 00000010 D	
FAB\$C_VAR	= 00000002	D	RAB\$W_RSZ	= 00000022 D	
FAB\$C_VFC	= 00000005	D	REC_OK	000006BE R D 01	
FAB\$L_ALQ	= 00000010	D	RELATIVE_FAB	00000000 RG D 01	
FAB\$L_DNA	= 00000030	D	RELATIVE_RAB	00000050 R D 01	
FAB\$L_FNA	= 0000002C	D	RELBSZ	000000C8 G D 01	
FAB\$L_FOP	= 00000004	D	RELBUF	000000F0 RG D 01	
FAB\$L_STV	= 0000000C	D	REL_TEST	0000032D R D 01	
FAB\$L_XAB	= 00000024	D	REL_TEST2	00000400 R D 01	
FAB\$M_CIF	= 02000000	D	REPORT_ERR	***** X 01	
FAB\$M_SUP	= 00000004	D	REPORT_ERROR	***** X 01	
FAB\$V_CHAN_MODE	= 00000002	D	RFATBL	000000BE R D 01	
FAB\$V_CR	= 00000001	D	RFA_OK	00000853 R D 01	
FAB\$V_DEL	= 00000002	D	RFMC	00000029 R D 01	
FAB\$V_DFW	= 00000005	D	RFMS	000001C0 R D 01	

```

RFMSTR
RMSS_CREATED
RMSS_EOF
RMSS_RNF
RMSTEST_3A
RMSTEST_5A
RNOK
RSZ_OK
SAV
SAV_RFA
SETOP
STVOK
SYSSCLOSE
SYSSCONNECT
SYSSCREATE
SYSSDELETE
SYSSDISCONNECT
SYSEXEND
SYSSFIND
SYSSGET
SYSSOPEN
SYSSPUT
SYSSREWIND
SYSSUPDATE
T3START
UPDATE_RECORD
VERBOSITY
XAB$B_AID
XAB$B_AOP
XAB$B_BKZ
XAB$C_ALL
XAB$C_ALLLEN
XAB$L_ALQ
XAB$L_LOC
XAB$L_NXT
XAB$W_DEQ
XAB$W_RFI0
XAB$W_RFI2
XAB$W_RFI4
XAB$W_VOL
          000001B8 R D 01
          ***** X 01
          ***** X 01
          ***** X 01
          000001E9 RG D 01
          ***** X 01
          0000081A R D 01
          000006A0 R D 01
          00000859 R D 01
          0000082C R D 01
          00000787 R D 01
          00000389 R D 01
          ***** GX 01
          00000000 RG D 01
          0000058A R D 01
          ***** X 01
          = 00000017 D
          = 00000008 D
          = 00000016 D
          = 00000014 D
          = 00000020 D
          = 00000010 D
          = 0000000C D
          = 00000004 D
          = 00000014 D
          = 00000018 D
          = 0000001A D
          = 0000001C D
          = 0000000A D

```

+-----+
! Psect synopsis !
+-----+

PSECT name

	Allocation	PSECT No.	Attributes
· ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
· RMSTEST	000008C8 (2248.)	01 (1.)	NOPIC USR CON REL GBL NOSHR EXE RD WRT NOVEC LONG
· XAB\$B	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
· SRMSNAM	0000000F (15.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
__SRMSNAM	00000194 (404.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	38	00:00:00.09	00:00:00.38
Command processing	133	00:00:00.60	00:00:03.73
Pass 1	299	00:00:11.36	00:00:25.25
Symbol table sort	0	00:00:00.55	00:00:01.11
Pass 2	118	00:00:02.85	00:00:06.66
Symbol table output	19	00:00:00.12	00:00:00.16
Psect synopsis output	2	00:00:00.04	00:00:00.04
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	611	00:00:15.61	00:00:37.33

The working set limit was 1500 pages.

54514 bytes (107 pages) of virtual memory were used to buffer the intermediate code.

There were 30 pages of symbol table space allocated to hold 441 non-local and 20 local symbols.

665 source lines were read in Pass 1, producing 46 object records in Pass 2.

64 pages of virtual memory were used to define 49 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	35
TOTALS (all libraries)	35

870 GETS were required to define 35 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LI\$:\$:RMSTEST3/OBJ=OBJ\$:\$:RMSTEST3 MSRC\$:\$:RMSTEST3/UPDATE=(ENH\$:\$:RMSTEST3)+EXECMLS/LIB

0409 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

